

Purdue University

**Purdue e-Pubs**

---

Department of Computer Science Technical  
Reports

Department of Computer Science

---

1990

## Curvature Computations on Surfaces in n-Space

Jung-Hong Chuang

Christoph M. Hoffmann

*Purdue University*, [cmh@cs.purdue.edu](mailto:cmh@cs.purdue.edu)

Report Number:

90-1015

---

Chuang, Jung-Hong and Hoffmann, Christoph M., "Curvature Computations on Surfaces in n-Space" (1990). *Department of Computer Science Technical Reports*. Paper 17.  
<https://docs.lib.purdue.edu/cstech/17>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.  
Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

# **CURVATURE COMPUTATIONS ON SURFACES IN $n$ -SPACE\***

Jung-Hong Chuang<sup>†</sup>  
and  
Christoph M. Hoffmann<sup>‡</sup>

Computer Sciences Department  
Purdue University  
Technical Report CSD-TR-1015  
CAPO Report CER-90-34  
September, 1990

---

\* This research was supported, through the Leonardo Fibonacci Institute, by the Instituto Trentino di Cultura, Trento, Italy.

<sup>†</sup> Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China. Supported by the National Science Foundation under grant CCR 86-19817, and by the Office of Naval Research under contracts N00014-86-K-0465 and N00014-90-J-1599.

<sup>‡</sup> Department of Computer Science, Purdue University, West Lafayette, IN 47907. Supported by the National Science Foundation under grant CCR 86-19817, and by the Office of Naval Research under contracts N00014-86-K-0465 and N00014-90-J-1599.

# Curvature Computations on Surfaces in $n$ -Space\*

Jung-Hong Chuang<sup>†</sup>      Christoph M. Hoffmann<sup>‡</sup>

August 1990

## Abstract

Let  $\mathcal{F}$  be a two-dimensional manifold in  $n$ -dimensional space, and let  $\pi(\mathcal{F})$  be its projection into the subspace of three of the variables in which  $\mathcal{F}$  has been expressed. We give an algorithm that computes the normal curvature of  $\pi(\mathcal{F})$  directly from the equations of  $\mathcal{F}$  without variable elimination. We also comment on applications in computer-aided geometric design (CAGD).

## 1 Introduction

Many surface operations in CAGD derive new surfaces from given ones subject to certain constraints. Examples include offset surfaces, where a distance

---

\*This research was supported, through the Leonardo Fibonacci Institute, by the Istituto Trentino di Cultura, Trento, Italy.

<sup>†</sup>Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China. Supported by the National Science Foundation under Grant CCR 86-19817, and by the Office of Naval Research under contracts N00014-86-K-0465 and N00014-90-J-1599.

<sup>‡</sup>Department of Computer Science, Purdue University, West Lafayette, Indian 47907. Supported in part by the National Science Foundation under Grant CCR 86-19817, and by the Office of Naval Research under contracts N00014-86-K-0465 and N00014-90-J-1599.

constraint must be observed, and spherical blends, where a curvature and a continuity constraint must be satisfied. While there exist intuitive descriptions of the resulting surfaces that are easily grasped, a precise mathematical representation of the surfaces appears difficult to obtain in practice.

Indeed, closed-form representations, are available in principle with the help of elimination theory — e.g., [4, 5] — or using Gröbner bases techniques — e.g., [2, 3, 14]. In practice, however, closed forms are usually unobtainable because the elimination problems that must be solved are well beyond the capabilities of machines and algorithms available to date; see, e.g., [16]. In consequence, surface operations including offsets and blends have been treated individually in the literature, and specific approximation methods for the resulting surfaces have been given that are not general. For example, offset computations are addressed in [7, 10, 12, 13, 18, 19, 22, 25, 26], and many ingenious and useful techniques for analyzing and approximating offsets have been derived. But the methods proposed in those papers for deriving and analyzing offsets do not apply unchanged to, say, the treatment of other surfaces whose definition also involves distance constraints, such as Voronoi surfaces [6, 11].

In [6, 8, 11, 15, 16] we have given a uniform method for deriving an exact representation of offsets, blends, equal-distance surfaces, and so on. The representation is a system  $F = 0$  of nonlinear equations in  $n$  variables, where  $n > 3$ , with the property that the surface of interest is the natural projection of the solution set of the system into the subspace spanned by the first three variables. Thus, the surface is conceptualized as the projection of a certain 2-surface in  $n$ -space, and, as discussed in [6, 15, 16], the auxiliary variables in the system  $F = 0$  have a concrete geometric meaning that can be exploited for instance in engineering design applications.

In [14, 15], a uniform method for evaluating the intersection of such surfaces has been described. In this paper, we develop a uniform method for determining the surface curvature at a given point. That is, we present an algorithm for the following problem: A surface  $\mathcal{F}$  is given as a system of  $m$

nonlinear equations in  $n$  variables

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \dots, x_n) &= 0 \end{aligned}$$

It is assumed that the solution set of the system is locally a smooth 2-manifold in  $\mathbf{R}^n$ , and so we have ordinarily  $m = n - 2$ ; however, as discussed in [16], in certain situations  $m > n - 2$  is desirable. Let  $\pi(\mathcal{F})$  be the natural projection of  $\mathcal{F}$  into the  $(x_1, x_2, x_3)$ -subspace,  $p$  a point on  $\mathcal{F}$ , and  $\mathbf{v}$  a tangent direction to  $\mathcal{F}$  at the point  $p$ . Determine the normal curvature of  $\pi(\mathcal{F})$  at  $\pi(p)$  in the direction  $\pi(\mathbf{v})$ .

If the nonlinear equations  $f_k$  are algebraic, our problem can be solved in principle by eliminating  $x_4, \dots, x_n$  from the equation system, followed by the well-known curvature computation in 3-space using the *shape operator*; see, e.g., [21]. Such a solution would not be practical, however, for in CAGD applications the elimination of the auxiliary variables  $x_4, \dots, x_n$  usually cannot be carried out in practice. Hence, an algorithm is needed that avoids elimination altogether. Such an algorithm is given here.

## 2 An Example from Surface Blending

To illustrate how surfaces can be described as sets of nonlinear equations, we derive a ruled surface  $K$  that arises in certain approaches to blending two surfaces. e.g., [20, 23]. The approach can be conceptualized as follows: We are given two surfaces (or patches of surfaces)  $f$  and  $g$ , on which two link curves  $C_1$  and  $C_2$  have been specified. Note that  $f$  and  $g$  are usually assumed to be parametric, but this assumption is inessential and can be dropped. The blending surface  $h$  should be tangent to  $f$  in the curve  $C_1$ , and tangent to  $g$  in the curve  $C_2$ . See Figure 1 for an illustration.

Now, the approach is to put the points of  $C_1$  into 1-1 correspondence with the points of  $C_2$ , and to connect corresponding points  $p_1$  and  $p_2$  with a straight line  $L$ . Thereupon, a plane  $P$  through  $L$  is considered that intersects  $f$  and  $g$  in two curves,  $C_f$  and  $C_g$ , that are blended as curves by a curve  $C_h$

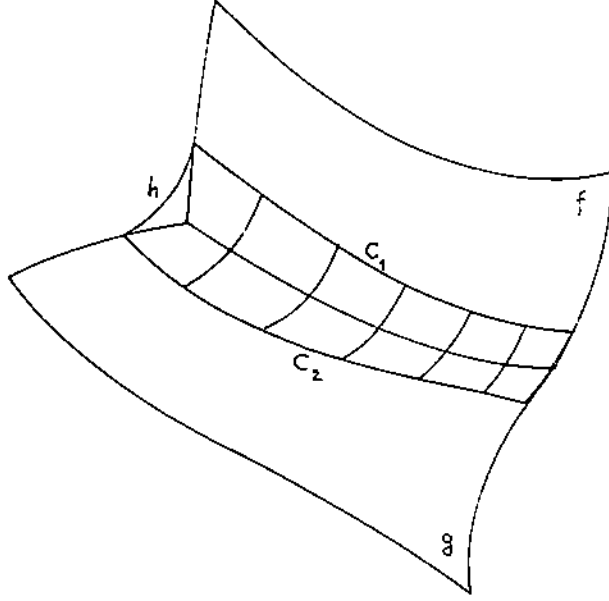


Figure 1: Blending Surface and Link Curves

that depends on  $L$  and on the way in which  $P$  has been chosen. Note that  $C_h$  can be obtained as a functional blend [20] or as a parametric curve [23]. See also Figure 2. The plane  $P$  can be suitably chosen, say by requiring that it is normal to  $f$  at  $p_1$ , as illustrated in Figure 3. The approach effectively reduces the three-dimensional surface-blending problem to a two-dimensional curve-blending problem, for it can then be proved that the surface obtained in this manner is a blending surface for  $f$  and  $g$ . Moreover, [24] proves that curvature continuity of the curve blend  $C_h$  implies curvature continuity of the surface  $h$  with  $f$ , provided that the tangents to  $C_h$  and  $C_1$  do not coincide at

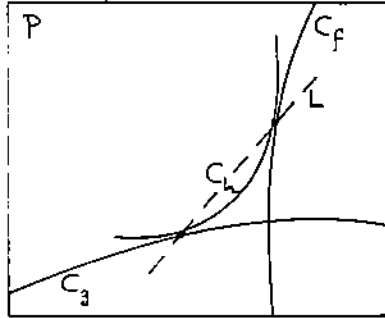


Figure 2: Blending Curve  $C_h$  in the Plane  $P$

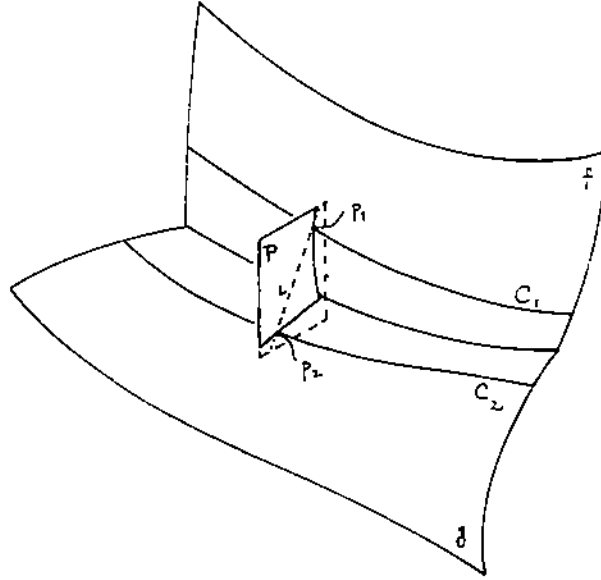


Figure 3: The Plane  $P$  through the Line  $L$

$p_1$ . That is, the planes  $P$  must not contain the tangents of the link curves.

A difficulty with this approach to constructing a blending surface is to find a simple method for establishing the correspondence of points on the link curves  $C_1$  and  $C_2$ . In [23], Pegna proposes the following idea: Design a space curve  $C_0$ , say as a Bezier curve, and let the link curves be its orthogonal projection onto the two surfaces. That is,  $C_1$  is the orthogonal projection of  $C_0$  onto  $f$ , whereas  $C_2$  is the orthogonal projection of  $C_0$  onto  $g$ . Then two points  $p_1$  and  $p_2$  on  $C_1$  and  $C_2$  correspond precisely when they are the image of the same point  $p_0$  on  $C_0$ . See also Figure 4.

The lines  $L$  under this point correspondence define a ruled surface  $K$ , and we describe a representation of  $K$  as a system of nonlinear equations. This is an example of the surfaces considered in this paper, and the derivation illustrates our methodology for deriving complex surface representations by devising systems of equations with auxiliary variables. Note that the methodology is akin to the equational programming paradigm in programming languages, e.g., [17], except that here the equations must be interpreted over a field whereas in programming language research the equations are typically understood over a free algebra.

We begin the surface derivation by assigning variable names to generic

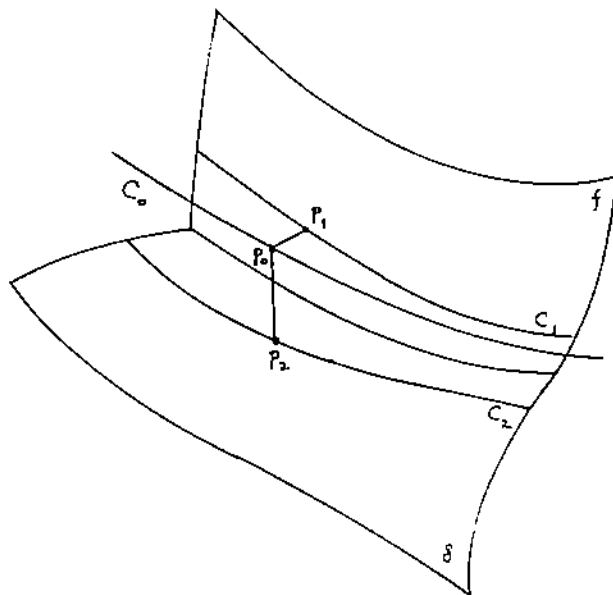


Figure 4: Point Correspondence on  $C_1$  and  $C_2$

point coordinates. Let

$$\begin{aligned}
 p_0 &= (u_0, v_0, w_0) && \text{be a generic point on } C_0, \\
 p_1 &= (u_1, v_1, w_1) && \text{the projection of } p_0 \text{ onto } f, \\
 p_2 &= (u_2, v_2, w_2) && \text{the projection of } p_0 \text{ onto } g, \text{ and let} \\
 q &= (x, y, z) && \text{be a generic point on the line } L \\
 &&& \text{through } p_1 \text{ and } p_2.
 \end{aligned}$$

Assume further that the curve  $C_0$  is given parametrically with the coordinate functions

$$C_0 : (H_1(r), H_2(r), H_3(r))$$

The surfaces  $f$  and  $g$  may be given parametrically or implicitly. For the sake of illustrating both cases, we will assume that  $f$  is given implicitly, as

$$f(x, y, z) = 0$$

and that  $g$  is given parametrically, by

$$g : (g_1(s, t), g_2(s, t), g_3(s, t))$$

It will then be clear how to modify the equations in case both surfaces are parametric or both are implicit.



We now compile the equations defining the system  $K$  by translating all geometric constraints on the points into equations. For each constraint, we obtain one or more equations as follows:

The point $p_0$ is on $C_0$ :	$u_0 = H_1(r)$ $v_0 = H_2(r)$ $w_0 = H_3(r)$
The point $p_1$ is on $f$ :	$f(u_1, v_1, w_1) = 0$
The point $p_2$ is on $g$ :	$u_2 = g_1(s, t)$ $v_2 = g_2(s, t)$ $w_2 = g_3(s, t)$
The point $q$ lies on the line $L$ , and $L$ contains $p_1$ and $p_2$ :	$x = \lambda u_1 + (1 - \lambda)u_2$ $y = \lambda v_1 + (1 - \lambda)v_2$ $z = \lambda w_1 + (1 - \lambda)w_2$
$p_1$ is the projection of $p_0$ onto $f$ :	$(u_0 - u_1, v_0 - v_1, w_0 - w_1) \cdot (0, -\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial v_1}) = 0$ $(u_0 - u_1, v_0 - v_1, w_0 - w_1) \cdot (\frac{\partial f}{\partial w_1}, 0, -\frac{\partial f}{\partial u_1}) = 0$ $(u_0 - u_1, v_0 - v_1, w_0 - w_1) \cdot (-\frac{\partial f}{\partial v_1}, \frac{\partial f}{\partial u_1}, 0) = 0$
$p_2$ is the projection of $p_0$ onto $g$ :	$(u_0 - u_2, v_0 - v_2, w_0 - w_2) \cdot (\frac{\partial g_1}{\partial s}, \frac{\partial g_2}{\partial s}, \frac{\partial g_3}{\partial s}) = 0$ $(u_0 - u_2, v_0 - v_2, w_0 - w_2) \cdot (\frac{\partial g_1}{\partial t}, \frac{\partial g_2}{\partial t}, \frac{\partial g_3}{\partial t}) = 0$

So, the ruled surface  $K$  with points  $q = (x, y, z)$  has been represented by 15 equations in the 16 variables

$$x, y, z, u_0, v_0, w_0, u_1, v_1, w_1, u_2, v_2, w_2, r, s, t, \lambda$$

For an example, see the appendix.

There is a redundancy in the equation system that has been introduced on purpose. when expressing the constraint that  $p_1$  is the orthogonal projection

onto the implicit surface  $f$ . Here

$$\begin{aligned} \mathbf{t}_1 &= (0, -\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial v_1}) \\ \mathbf{t}_2 &= (\frac{\partial f}{\partial w_1}, 0, -\frac{\partial f}{\partial u_1}) \\ \mathbf{t}_3 &= (-\frac{\partial f}{\partial v_1}, \frac{\partial f}{\partial u_1}, 0) \end{aligned}$$

are three tangent vectors to  $f$  at the point  $p_1$ , which is evident when the inner product with the gradient vector

$$\nabla f = (\frac{\partial f}{\partial u_1}, \frac{\partial f}{\partial v_1}, \frac{\partial f}{\partial w_1})$$

is computed. Clearly, there can be only two linearly independent vectors among the  $\mathbf{t}_k$ , but since some of the partial derivatives could vanish at  $p_1$ , we cannot decide *a-priori* which ones are independent. Instead, it is convenient to have an algebraic dependence in the system and adjust the algorithms that work with the system accordingly. See also [16].

### 3 Curvature Computations

We assume given a point  $p = (p_1, \dots, p_n)$  that satisfies the system  $F = 0$  of  $m$  equations in  $n$  variables:

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \dots, x_n) &= 0 \end{aligned}$$

Moreover, we assume that the hypersurfaces  $f_k = 0$  are smooth at  $p$ , and that the Jacobian of the system,

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

has rank  $n - 2$ , so that the solution set defined by  $F = 0$  is locally a two-dimensional manifold.

Let  $\pi$  be the natural projection function from  $\mathbb{R}^n$  to  $\mathbb{R}^3$  that maps points  $q = (x_1, \dots, x_n)$  to points  $\pi(q) = (x_1, x_2, x_3)$ . If  $\mathcal{F}$  is the solution of the system  $F = 0$  in  $\mathbb{R}^n$  and  $\pi(\mathcal{F})$  its projection into  $\mathbb{R}^3$ , then we will show that the normal curvature of  $\pi(\mathcal{F})$  at  $\pi(p)$  is determined by a certain linear combination of the second fundamental forms of the hypersurfaces  $f_k$ , where  $0 \leq k \leq m$ . Moreover, we will show how to compute this linear combination. In consequence, we can determine at  $\pi(p)$  all curvature properties of  $\pi(\mathcal{F})$  without the need to determine  $\pi(\mathcal{F})$  explicitly. The following lemma explains how to determine the normal to  $\pi(\mathcal{F})$  at the point  $\pi(p)$ , in  $\mathbb{R}^3$ .

**Lemma 1** Let  $\mathbf{n}_k$  be the normal vector to the hypersurface  $f_k$  at  $p \in \mathbb{R}^n$ , for  $1 \leq k \leq m$ . Let  $\alpha_k$  be such that the last  $n - 3$  components of

$$\mathbf{n}_0 = \sum_{k=1}^m \alpha_k \mathbf{n}_k = (a, b, c, 0, \dots, 0)$$

are zero. Then  $\pi(\mathbf{n}_0)$  is normal to  $\pi(\mathcal{F})$  at  $\pi(p)$ .

**Proof** If  $\mathbf{t} = (t_1, t_2, t_3, \dots)$  is tangent to  $\mathcal{F}$  at  $p$ , then  $\pi(\mathbf{t}) = (t_1, t_2, t_3)$  is tangent to  $\pi(\mathcal{F})$  at  $\pi(p)$ . Since  $\pi(\mathbf{t}) \cdot \pi(\mathbf{n}_0) = 0$ ,  $\mathbf{n}_0$  is normal to  $\pi(\mathcal{F})$  at  $\pi(p)$ .  $\square$

Meusnier's theorem in differential geometry states that the second derivative of any curve, through a point  $p$  that lies on a smooth surface  $f$  in 3-space, has a projection onto the surface normal that does not depend on the curve. The theorem generalizes to higher dimensions, where it can be stated as follows:

**Lemma 2 (Meusnier)** Let  $\beta(t)$  be a curve on the hypersurface  $f$  in  $\mathbb{R}^n$  through the point  $p = \beta(0)$ , and let  $\mathbf{v} = \partial\beta(0)/\partial t$ . Then

$$L_p(\mathbf{v}) \cdot \mathbf{v} = -\nabla_{\mathbf{v}} N = \frac{\partial^2 \beta(0)}{\partial t^2} \cdot \mathbf{n}$$

where  $\mathbf{n}$  is the normal to  $f$  at  $p$ , and  $N$  is the (unit) normal vector field of  $f$ .

Note that  $L_p(\mathbf{v})$  is the shape operator. Therefore, we can define the normal curvature of the hypersurface  $f$  at  $p$  in the tangent direction  $\mathbf{v}$  as

$$\kappa(\mathbf{v}) = L_p(\mathbf{v}) \cdot \mathbf{v}$$

where  $\mathbf{v}$  has unit length. We now develop the main theorem.

Consider a curve  $\beta(t)$  on the 2-surface  $\mathcal{F}$  in  $\mathbf{R}^n$  that contains  $p$ , and assume that

$$\begin{aligned}\beta(0) &= p \\ \frac{\partial \beta(0)}{\partial t} &= \mathbf{v}\end{aligned}$$

Then the curve lies on every hypersurface  $f_k$ , for  $1 \leq k \leq m$ , and  $\mathbf{v}$  is tangent to  $\mathcal{F}$  and each  $f_k$ .

The curve  $\beta$  projects to

$$\pi(\beta(t)) = (\beta_1(t), \beta_2(t), \beta_3(t))$$

We choose numbers  $\alpha_k$  such that

$$\mathbf{n}_0 = \sum_{k=1}^m \alpha_k \mathbf{n}_k = (a, b, c, 0, \dots, 0)$$

where the  $\mathbf{n}_k$  are normals to  $f_k$  at  $p$ , and  $a^2 + b^2 + c^2 = 1$ . That is, the projected normal  $\pi(\mathbf{n}_0)$  has unit length. We will abbreviate  $\frac{\partial \beta}{\partial t}$  with  $\dot{\beta}$  and  $\frac{\partial^2 \beta}{\partial t^2}$  with  $\ddot{\beta}$ . With  $L_p^{(k)}$  the shape operator of  $f_k$  at  $p$ , we obtain

$$\begin{aligned}\sum_{k=1}^m \alpha_k (L_p^{(k)}(\mathbf{v}) \cdot \mathbf{v}) &= \sum_{k=1}^m \alpha_k (\ddot{\beta}(0) \cdot \mathbf{n}_k) \\ &= \sum_{k=1}^m (\ddot{\beta}(0) \cdot \alpha_k \mathbf{n}_k) \\ &= \ddot{\beta}(0) \cdot (\sum_{k=1}^m \alpha_k \mathbf{n}_k) \\ &= \ddot{\beta}(0) \cdot (a, b, c, 0, \dots, 0) \\ &= (\ddot{\beta}_1(0), \ddot{\beta}_2(0), \ddot{\beta}_3(0)) \cdot \pi(\mathbf{n}_0)\end{aligned}\tag{1}$$

So, if the curve  $\beta(t)$  has been parameterized such that the vector

$$\pi(\mathbf{v}) = (\dot{\beta}_1(0), \dot{\beta}_2(0), \dot{\beta}_3(0))$$

has unit length, then expression (1) is the normal curvature of  $\pi(\mathcal{F})$  at  $\pi(p)$  in the direction  $\pi(\mathbf{v})$ . We summarize this fact as follows:

**Theorem 3** Let  $p$  be a point on the 2-surface  $\mathcal{F}$  in  $\mathbf{R}^n$ , where  $\mathcal{F}$  is the intersection of  $m$  smooth hypersurfaces  $f_k$ , and assume that  $\mathcal{F}$  is smooth at  $p$ . Let  $\mathbf{v} = (v_1, \dots, v_m)$  be a tangent vector to  $\mathcal{F}$  at  $p$ , with  $v_1^2 + v_2^2 + v_3^2 = 1$ ,

and let there be numbers  $\alpha_k$  such that  $\mathbf{n}_0 = \sum_{k=1}^m \alpha_k \mathbf{n}_k$ , where  $\pi(\mathbf{n}_0)$  has unit length and its last  $n - 3$  components are zero. If  $L_p^{(k)}$  is the shape operator of the hypersurface  $f_k$  at  $p$ , then

$$\kappa = \sum_{k=1}^m \alpha_k L_p^{(k)}(\mathbf{v}) \cdot \mathbf{v}$$

is the normal curvature of  $\pi(\mathcal{F})$  at  $\pi(p)$  in the direction  $\pi(\mathbf{v})$ .

In consequence, the following algorithm computes the normal curvature of the projected surface  $\pi(\mathcal{F})$ :

1. Adjust the length of the tangent vector  $\mathbf{v}$  such that its projection  $\pi(\mathbf{v})$  has unit length.
2. For each hypersurface  $f_k$  compute the normal vector  $\mathbf{n}_k$  at  $p$ .
3. Find a unit normal to the projected surface by solving the linear system entailed by  $\sum_{k=1}^m \alpha_k \mathbf{n}_k = (a, b, c, 0, \dots, 0)$ , and then adjusting the  $\alpha_k$  such that  $a^2 + b^2 + c^2 = 1$ .
4. For  $1 \leq k \leq m$ , compute  $L_p^{(k)}(\mathbf{v}) \cdot \mathbf{v}$ .
5. Compute  $\kappa = (\sum_{k=1}^m \alpha_k L_p^{(k)}(\mathbf{v})) \cdot \mathbf{v}$ .

It is straightforward to implement this method. A different algorithm for computing the curvature of the projected surface can be given also, based on finding a parametric curve  $\beta(t)$  on  $\mathcal{F}$ , see [8].

Note that the computation of  $L_p^{(k)}(\mathbf{v}) \cdot \mathbf{v}$  can be based on the following observation.

**Lemma 4** Let  $g = 0$  be a hypersurface, and let  $N = \nabla g$  be the normal vector field of  $g$  at the point  $p$ . With  $H$  the Hessian matrix of  $g$  at  $p$ , if  $\mathbf{v} = (v_1, \dots, v_n)$  is a tangent vector of  $g = 0$  at  $p$ , then  $L_p(\mathbf{v}) \cdot \mathbf{v} = -\mathbf{v}^T H \mathbf{v}$

**Proof**

$$\begin{aligned} L_p(\mathbf{v}) \cdot \mathbf{v} &= -\nabla_{\mathbf{v}} N \cdot \mathbf{v} \\ &= -\nabla_{\mathbf{v}} \nabla g \cdot \mathbf{v} \end{aligned}$$

$$\begin{aligned}
&= - \left( \nabla_{\mathbf{v}} \frac{\partial g}{\partial x_1}, \nabla_{\mathbf{v}} \frac{\partial g}{\partial x_2}, \dots, \nabla_{\mathbf{v}} \frac{\partial g}{\partial x_n} \right) \cdot \mathbf{v} \\
&= - \left[ \nabla \left( \frac{\partial g}{\partial x_1} \right) (p) \cdot \mathbf{v}, \nabla \left( \frac{\partial g}{\partial x_2} \right) (p) \cdot \mathbf{v}, \dots, \nabla \left( \frac{\partial g}{\partial x_n} \right) (p) \cdot \mathbf{v} \right] \cdot \mathbf{v} \\
&= - \left[ \sum_{i=1}^n \frac{\partial^2 g}{\partial x_i \partial x_1} (p) v_i, \sum_{i=1}^n \frac{\partial^2 g}{\partial x_i \partial x_2} (p) v_i, \dots, \sum_{i=1}^n \frac{\partial^2 g}{\partial x_i \partial x_n} (p) v_i \right] \cdot \mathbf{v} \\
&= - \sum_{i,j=1}^n \frac{\partial^2 g}{\partial x_i \partial x_j} (p) v_i v_j \\
&= -\mathbf{v}^T H \mathbf{v}
\end{aligned}$$

□

Because of the bilinearity of the form  $-\mathbf{v}^T H \mathbf{v}$ , notice that we can rephrase Theorem 3 in a form better suited to computing the normal curvature at  $\pi(p)$  in different directions.

**Corollary 5** Let  $p$  be a point on the 2-surface  $\mathcal{F}$  in  $\mathbb{R}^n$ , where  $\mathcal{F}$  is the intersection of  $m$  smooth hypersurfaces  $f_k$ , and assume that  $\mathcal{F}$  is smooth at  $p$ . Let  $\mathbf{v} = (v_1, \dots, v_m)$  be a tangent vector to  $\mathcal{F}$  at  $p$ , with  $v_1^2 + v_2^2 + v_3^2 = 1$ , and let there be numbers  $\alpha_k$  such that  $\mathbf{n}_0 = \sum_{k=1}^m \alpha_k \mathbf{n}_k$  satisfies the hypotheses of Theorem 3. With  $\mathbf{n}_k = \nabla f_k$  and  $H_k$  the Hessian of the hypersurface  $f_k$  at  $p$ , let

$$H_0 = \sum_{k=1}^m \alpha_k H_k$$

Then

$$\kappa = -\mathbf{v}^T H_0 \mathbf{v}$$

is the normal curvature of  $\pi(\mathcal{F})$  at  $\pi(p)$  in the direction  $\pi(\mathbf{v})$ .

## 4 Normal Curvature of a Parametric Surface

We illustrate our results by deriving a formula for the normal direction and for the normal curvature of parametric surfaces, in the direction of the isoparametric lines. Another example is given in the appendix.

We consider a parametric surface

$$\begin{aligned}x &= h1(s, t) \\y &= h2(s, t) \\z &= h3(s, t)\end{aligned}$$

as the projection, into  $(x, y, z)$ -space, of the 2-surface in 5-dimensional  $(x, y, z, s, t)$ -space obtained by intersecting the three hypersurfaces

$$\begin{aligned}f_1 : x - h1(s, t) \\f_2 : y - h2(s, t) \\f_3 : z - h3(s, t)\end{aligned} \tag{2}$$

The coordinate functions  $h1, h2, h3$  are assumed to be analytic and twice differentiable. This assumption holds in particular for the rational polynomial functions used in CAGD. We abbreviate the partial derivatives of the coordinate functions by subscripts; for example, we write  $h2_s$ , instead of  $\partial h2 / \partial s$ .

The normals to the  $f_k$  are

$$\begin{aligned}\nabla f_1 &= (1, 0, 0, -h1_s, -h1_t) \\ \nabla f_2 &= (0, 1, 0, -h2_s, -h2_t) \\ \nabla f_3 &= (0, 0, 1, -h3_s, -h3_t)\end{aligned} \tag{3}$$

Applying Lemma 1, we must solve a linear system in order to obtain the normal direction to the parametric surface:

$$\begin{aligned}\alpha_1 h1_s + \alpha_2 h2_s + \alpha_3 h3_s &= 0 \\ \alpha_1 h1_t + \alpha_2 h2_t + \alpha_3 h3_t &= 0\end{aligned} \tag{4}$$

The system is solved by

$$\begin{aligned}\alpha_1 &= h2_s h3_t - h3_s h2_t \\ \alpha_2 &= h3_s h1_t - h1_s h3_t \\ \alpha_3 &= h1_s h2_t - h2_s h1_t\end{aligned} \tag{5}$$

In consequence, the normal vector to the parametric surface is given by

$$(\alpha_1, \alpha_2, \alpha_3) = (h1_s, h2_s, h3_s) \times (h1_t, h2_t, h3_t)$$

To obtain the unit normal  $\mathbf{n}_0$  in projection, we must adjust the  $\alpha_k$  by dividing by the length of the cross product, i.e.,

$$\begin{aligned}\mathbf{n}_0 &= (\mathbf{h}_{1s}, \mathbf{h}_{2s}, \mathbf{h}_{3s}) \times (\mathbf{h}_{1t}, \mathbf{h}_{2t}, \mathbf{h}_{3t})/m \\ m &= \|(\mathbf{h}_{1s}, \mathbf{h}_{2s}, \mathbf{h}_{3s}) \times (\mathbf{h}_{1t}, \mathbf{h}_{2t}, \mathbf{h}_{3t})\|\end{aligned}$$

Next, we compute the normal curvature in the tangent direction of the isoparametric line  $t = \text{const.}$  The Hessians of the system (2) are the matrices

$$\begin{aligned}H_1 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_{1ss} & -h_{1st} \\ 0 & 0 & 0 & -h_{1st} & -h_{1tt} \end{pmatrix} \\ H_2 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_{2ss} & -h_{2st} \\ 0 & 0 & 0 & -h_{2st} & -h_{2tt} \end{pmatrix} \\ H_3 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_{3ss} & -h_{3st} \\ 0 & 0 & 0 & -h_{3st} & -h_{3tt} \end{pmatrix}\end{aligned}$$

Therefore, the matrix  $H_0$  is given by

$$H_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a & b \\ 0 & 0 & 0 & b & c \end{pmatrix} \quad (6)$$

where

$$\begin{aligned}a &= -(\mathbf{h}_{1ss}, \mathbf{h}_{2ss}, \mathbf{h}_{3ss}) \cdot (\mathbf{h}_{1s}, \mathbf{h}_{2s}, \mathbf{h}_{3s}) \times (\mathbf{h}_{1t}, \mathbf{h}_{2t}, \mathbf{h}_{3t})/m \\ b &= -(\mathbf{h}_{1st}, \mathbf{h}_{2st}, \mathbf{h}_{3st}) \cdot (\mathbf{h}_{1s}, \mathbf{h}_{2s}, \mathbf{h}_{3s}) \times (\mathbf{h}_{1t}, \mathbf{h}_{2t}, \mathbf{h}_{3t})/m \\ c &= -(\mathbf{h}_{1tt}, \mathbf{h}_{2tt}, \mathbf{h}_{3tt}) \cdot (\mathbf{h}_{1s}, \mathbf{h}_{2s}, \mathbf{h}_{3s}) \times (\mathbf{h}_{1t}, \mathbf{h}_{2t}, \mathbf{h}_{3t})/m \\ m &= \|(\mathbf{h}_{1s}, \mathbf{h}_{2s}, \mathbf{h}_{3s}) \times (\mathbf{h}_{1t}, \mathbf{h}_{2t}, \mathbf{h}_{3t})\|\end{aligned}$$



Now the vector

$$\mathbf{v} = (h1_s, h2_s, h3_s, 1, 0)$$

is tangent to the 2-surface defined by (2) and projects to the tangent of the isoparametric line  $t = \text{const.}$  When divided by  $n = \|(h1_s, h2_s, h3_s)\|$ , the projected vector has unit length. In consequence, the curvature is given by

$$\begin{aligned}\kappa_{t=\text{const}} &= -\mathbf{v}^T H_0 \mathbf{v} / n^2 \\ &= -a / n^2\end{aligned}$$

Similarly, the normal curvature in the direction of the isoparametric line  $s = \text{const}$  is

$$\kappa_{s=\text{const}} = -c / n^2$$

## 5 Summary

We have presented a method for determining the local curvature of the projection, into 3-space, of a 2-surface in  $n$ -space. The method does not rely on expensive elimination computations, and we have implemented it. Our techniques are useful in situations in which complex surfaces cannot be represented in a simple closed form, or are easily approximated by parametric surfaces, when the surfaces instead have been expressed using systems of equations in more than three variables. Examples of such surfaces include offset surfaces, Voronoi surfaces, fixed and variable-radius spherical blending surfaces, and auxiliary surfaces such as the ruled surface  $K$  of Section 2 or the trimming surfaces [11] used in the definition of the skeleton. Algorithms such as the one presented here are part of an infrastructure of surface interrogation methods that should make surface representations by systems of equations a reasonable alternative allowing practical work with geometrically constrained surfaces.

## References

1. E.L. Allgower and S. Gnatzmann (1987), "An Algorithm for Piecewise Linear Approximation of Implicitly Defined 2-Dimensional Surfaces," *SIAM J. Num. Anal.* 24, 452-469.

2. B. Buchberger (1985), "Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory," in *Multidimensional Systems Theory*, N.K. Bose, ed., D. Reidel Publishing Co., Dordrecht, Holland; 184-232.
3. B. Buchberger, G. Collins, and B. Kutzler (1988), "Algebraic Methods for Geometric Reasoning," *Annl. Rev. in Comp. Science* 3, 85-120.
4. A. Cayley (1848), "On the Theory of Elimination." *Cambridge and Dublin Math. J.* 3, 116-120.
5. E.-W. Chionh (1990), "Base Points, Resultants, and the Implicit Representation of Rational Surfaces," PhD Diss., Comp. Sci., Univ. Waterloo, Canada.
6. V. Chandru, D. Dutta, C. Hoffmann (1990), "Variable Radius Blending with Cyclides," in *Geometric Modeling for Product Engineering*, K. Preiss, J. Turner, M. Wozny, eds., North Holland, 39-58.
7. J. Chou, E. Cohen (1989), "Constant Scallop Height Tool Path Generation," Rept. UUCS-89-011, Comp. Sci., Univ. Utah.
8. Jung-Hong Chuang (1990), *Surface Approximations in Geometric Modeling*, PhD Diss., Comp. Sci., Purdue University.
9. J.-H. Chuang and C.M. Hoffmann (1989), "On Local Implicit Approximation and Its Applications," *ACM Trans. Graphics* 8, 298-324.
10. S. Coquillart (1987), "Computing offsets of B-spline curves," *CAD* 19, 305-309.
11. D. Dutta and C. M. Hoffmann (1990), "A Geometric Investigation of the Skeleton of CSG Objects," Report CSD-TR-955, Comp. Sci., Purdue Univ.
12. R. Farouki (1986), "The Approximation of Nondegenerate Offset Surfaces," *Comp. Aided Geom. Design* 3, 15-43.
13. R. Farouki and C. Neff (1989), "Some Analytic and Algebraic Properties of Plane Offset Curves," Rept. RC 14364, IBM Yorktown Heights.

14. C. M. Hoffmann (1989), *Geometric and Solid Modeling, An Introduction*, Morgan Kaufmann Publ., San Mateo, Cal.
15. C. M. Hoffmann (1990), "A Dimensionality Paradigm for Surface Interrogation," to appear in *CAGD*.
16. C. M. Hoffmann (1990), "Algebraic and Numerical Techniques for Offsets and Blends," in *Computations of Curves and Surfaces*, W. Dahmen, M. Gasca, C. Micchelli, eds., Kluwer Acad. Publ., 499-528.
17. C. M. Hoffmann and M. J. O'Donnell (1982), "Programming with Equations," *ACM Trans Progr Lang* 4, 83-112.
18. J. Hoschek (1985), "Offset curves in the plane." *CAD* 17, 77-82.
19. J. Hoschek (1988), "Spline approximation of offset curves," *CAGD* 5, 33-40.
20. J. Li, J. Hoschek, E. Hartmann (1990), "A geometrical method for smooth joining and interpolation of curves and surfaces," *CAGD* 7.
21. B. O'Neill (1966), *Elementary Differential Geometry*, Academic Press.
22. B. Pham (1988), "Offset approximation of uniform B-splines," *CAD* 20, 471-474.
23. J. Pegna (1988), "Exact second order continuous interactive surface blending with variable sweep geometric modeling," *J. Offshore Mechanics and Arctic Engr.*
24. J. Pegna and F.-E. Wolter (1989), "A Simple Practical Criterion to Guarantee Second-Order Smoothness of Blend Surfaces," Proc. 1989 ASME Design Autom. Conf., Montreal, Canada.
25. J. Rossignac, A. Requicha (1986), "Offsetting operations in solid modeling," *CAGD* 3, 129-148.
26. S.E.O. Saeed, A. de Pennington, J.R. Dodsworth (1988), "Offsetting in geometric modeling," *CAD* 20, 67-74.

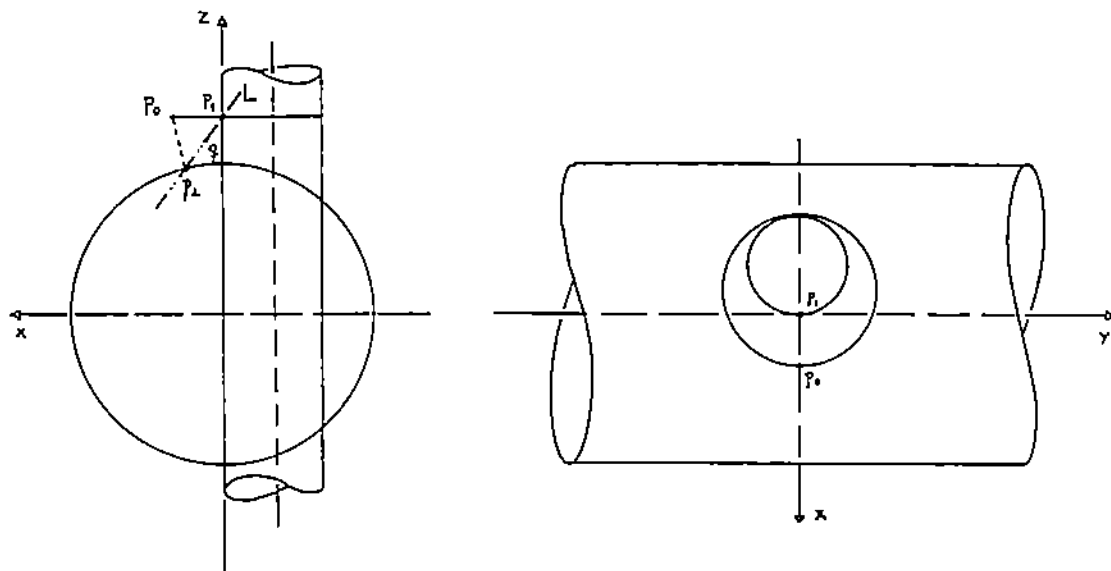


Figure 5: The surfaces  $f$  and  $g$ , the curve  $C_0$ , a generator of the ruled surface, and corresponding points.

## Appendix: Curvature of a Ruled Surface

We give an example of a ruled surface of the type described in Section 2 and compute its curvature in the direction of the ruling. Note that the curvature should then be zero.

Let  $f$  be a cylinder of radius 1 whose axis is parallel to the  $z$ -axis through the point  $(-1, 0, 0)$ . Let  $g$  be a cylinder of radius 3 whose axis is the  $y$ -axis. We assume that  $f$  is given implicitly and that  $g$  is given parametrically. Furthermore, let  $C_0$  be a circle in the plane  $z = 4$  of radius  $3/2$  centered at  $(-1/2, 0, 4)$ .  $C_0$  is also given parametrically. We formulate the equations of Section 2 that describe the ruled surface that passes through the two

orthogonal projections of  $C_0$  onto  $f$  and  $g$ .

$$\begin{aligned}
u_0 &= \frac{3(1-r^2)}{2(1+r^2)} - \frac{1}{2} \\
v_0 &= \frac{3r}{1+r^2} \\
w_0 &= 4 \\
(u_1+1)^2 + v_1^2 - 1 &= 0 \\
u_2 &= \frac{3(1-s^2)}{1+s^2} \\
v_2 &= t \\
w_2 &= \frac{6s}{1+s^2} \\
x &= \lambda u_1 + (1-\lambda)u_2 \\
y &= \lambda v_1 + (1-\lambda)v_2 \\
z &= \lambda w_1 + (1-\lambda)w_2 \\
2(w_0 - w_1)(u_1 + 1) &= 0 \\
-2(u_0 - u_1)v_1 + 2(v_0 - v_1)(u_1 + 1) &= 0 \\
2(w_0 - w_1)v_1 &= 0 \\
-12s(u_0 - u_2) + 6(1-s^2)(w_0 - w_2) &= 0 \\
v_0 - v_2 &= 0
\end{aligned}$$

Note that there are 16 variables and 15 equations.

On the surface defined by the equations, we choose the point

$$\begin{aligned}
\mathbf{p} &= (x, y, z, u_0, v_0, w_0, u_1, v_1, w_1, u_2, v_2, w_2, r, s, t, \lambda) \\
&= \left( \frac{3}{2\sqrt{17}}, 0, 2 + \frac{6}{\sqrt{17}}, 1, 0, 4, 0, 0, 4, \frac{3}{\sqrt{17}}, 0, \frac{12}{\sqrt{17}}, 0, \frac{\sqrt{17}-1}{4}, 0, \frac{1}{2} \right)
\end{aligned}$$

The curve and surfaces are shown in two projections in Figure 5. The points in the figure are the following projections of  $\mathbf{p}$ :

$$\begin{aligned}
q &= (x, y, z) \\
p_0 &= (u_0, v_0, w_0) \\
p_1 &= (u_1, v_1, w_1) \\
p_2 &= (u_2, v_2, w_2)
\end{aligned}$$

Note that  $q = \lambda p_1 + (1 - \lambda)p_2$ . For the point  $\mathbf{p}$  in  $\mathbf{R}^{16}$ , we construct the sum of the Hessians. The resulting matrix  $H_0$  is as follows

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\alpha_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & -\alpha_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\alpha_1 & \alpha_4 & 0 & \alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & -\alpha_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\alpha_6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_7 & 0 & \alpha_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_8 & 0 & \alpha_6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_9 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\alpha_2 & 0 & -\alpha_3 & 0 & 0 & 0 & \alpha_7 & 0 & \alpha_8 & 0 & \alpha_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\alpha_5 & 0 & -\alpha_6 & \alpha_5 & 0 & \alpha_6 & 0 & 0 & 0 & 0 \end{pmatrix}$$

where

$$\begin{aligned} \alpha_1 &= 0.2776737 & \alpha_2 &= 0.3038138 \\ \alpha_3 &= 0.2372107 & \alpha_4 &= 0.4158092 \\ \alpha_5 &= 0.8316184 & \alpha_6 &= 0.5553474 \\ \alpha_7 &= 0.4770799 & \alpha_8 &= 0.5641258 \\ \alpha_9 &= 1.4232639 & \alpha_0 &= 1.7147960 \end{aligned}$$

At the point  $\mathbf{p}$ , the tangent vector in 16-dimensional space is

$$\mathbf{v} = \left( \frac{3}{\sqrt{17}}, 0, \frac{12}{\sqrt{17}} - 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1 \right)$$

and projects in 3-space to the vector

$$\left( \frac{3}{\sqrt{17}}, 0, \frac{12}{\sqrt{17}} - 4 \right)$$

which is in the direction of the ruling. We compute

$$\mathbf{v} H_0 \mathbf{v}^T = 0$$

so the normal curvature of the projected surface, in the direction of the ruling, is zero.